CLAIMS

1.   (Currently Amended) A method of preventing system resource and command conflicts in a multiprocessor system, comprising the steps of:

issuing, by a given processor, a processor command, wherein the processor command is waiting to be selected for snooping;

comparing an address of the processor command [[s]] to addresses of all to prior snoop [[s]] commands occurring within a predetermined time window;

determining if the processor command and the prior snoop commands are non-pipelined; and

in response to the address of the processor command matching one or more of the addresses of the prior snoop commands occurring within the predetermined time window, or in response to determining that the processor command is non-pipelined and one or more of the prior snoop commands occurring within the predetermined time window are non-pipelined, delaying sourcing of the processor command until the end of a second predetermined time window.

~~detecting whether the addresses of these commands match said prior snoops [[']] coherency blocks; and~~

~~delaying the sourcing of the snoop commands which do match until the end of a specified window.~~

2.   (Currently Amended) The method of claim 1 wherein the comparing and detecting includes applying a given set of arbitration rules to ascertain potential resource and command conflicts.

3.   (Currently Amended) The method of claim 1 wherein the comparing, determining, and ~~snoop~~ sourcing occurs separately from any of the processors of the multiprocessor system.

4.   (Currently Amended) The method of claim 3 wherein the comparing, determining, and ~~snoop~~ sourcing occurs in an intelligent switch controller.

5.   (Original) A method of preventing system command conflicts in a multiprocessor system, comprising the steps of:

determining whether a command issued by a given processor is denoted to be a non-pipelined command;

detecting whether any prior commands issued within a specified time defined window are non-pipelined commands; and

delaying the sourcing of a snoop of any such non-pipelined command if there is any other comparable non-pipelined command in the specified window.

6.   (Original) The method of claim 5 wherein the determining and detecting and snoop sourcing occurs separately from any of the processors of the multiprocessor system.

7.   (Original) The method of claim 6 wherein the comparing and snoop sourcing occurs in an intelligent switch controller.

8.   (Original) A method of preventing system resource and command conflicts in a multiprocessor system, comprising the steps of:

comparing processor commands to prior snoops within a predetermined time defined window;

detecting whether the addresses of these commands match coherency blocks of said prior snoops;

determining whether a command issued by a given processor is denoted to be a non-pipelined command;

detecting whether any prior commands issued within a specified time defined window are non-pipelined commands where said specified and predetermined time defined window may be identical;

delaying the sourcing of the snoop commands which do match until after the end of the predetermined time defined window; and

delaying the sourcing of the snoop of any such non-pipelined command if there is any other comparable non-pipelined command in the specified time defined window.

9.   (Currently Amended) Apparatus for preventing system resource <u>and command</u> conflicts in a multiprocessor system, comprising:

a plurality of processors;

an intelligent switch connected to each of said plurality of processors for routing commands received from said processors;

means for comparing processor commands to prior snoops ~~within a predetermined time window~~;

<u>means for determining whether a command issued by a given processor is denoted to be a non-pipelined command;</u>

means for detecting <u>within a predetermined time window</u> whether the addresses of these commands match said prior snoops coherency blocks <u>and if any prior commands issued are non-pipelined commands</u>; and

means for delaying the sourcing of the snoop of commands which do match and <u>the non-pipelined commands if there is any comparable non-pipelined command</u> until the end of <u>a</u> ~~the~~ specified window.

10.   (Currently Amended) The apparatus of claim 9 wherein a given set of arbitration rules is used by said intelligent switch to ascertain potential resource <u>and command</u> conflicts.

11.   (Currently Amended) The apparatus of claim 9 wherein the comparing<u>, determining</u> and ~~snoop~~ sourcing occurs separately from any of the processors of the multiprocessor system.

12.   (Original) Apparatus for preventing system command conflicts in a multiprocessor system, comprising:

a plurality of processors;

an intelligent switch connected to each of said plurality of processors for routing commands received from said processors;

means for determining whether a command issued by a given processor is denoted to be a non-pipelined command;

means for detecting whether any prior commands issued within a specified time defined window are non-pipelined commands; and

means for delaying the sourcing of a snoop of any such non-pipelined command if there is any other comparable non-pipelined command in the specified window.

13.   (Original) The apparatus of claim 12 wherein said determining and snoop sourcing occurs separately from any of the processors of the multiprocessor system.

14.   (Original) The apparatus of claim 12 wherein said intelligent switch performs said determining and snoop sourcing.

15.   (Original) Apparatus for preventing system resource and command conflicts in a multiprocessor system, comprising:

means for comparing processor commands to prior snoops within a predetermined time defined window;

means for detecting whether the addresses of these commands match coherency blocks of said prior snoops;

means for determining whether a command issued by a given processor is denoted to be a non-pipelined command;

means for detecting whether any prior commands issued within a specified time defined window are non-pipelined commands where said specified and predetermined time defined window may be identical;

means for delaying the sourcing of the snoop commands which do match until after the end of the predetermined time defined window; and

means for delaying the sourcing of the snoop of any such non-pipelined command if there is any other comparable non-pipelined command in the specified time defined window.

16.   (New) The method of claim 1 wherein in response to the address of the processor command not matching the addresses of the prior snoop commands occurring within the predetermined time window, and in response to determining that the processor command is

either pipelined, or non-pipelined and none of the prior snoop commands occurring within the predetermined time window are non-pipelined, sourcing the processor command without substantial delay.

17. (New) The method of claim 1 wherein the predetermined time window and the second predetermined time window are identical.